P14.5: Write a telephone lookup program. Read a data set of 1,000 names and telephone

numbers from a file that contains the numbers in random order. Handle lookups

by name and also reverse lookups by phone number. Use a binary search for both

lookups.

P14.19: Supply a class ***Person*** that implements the ***Comparable*** interface. Compare persons by their names (last, first). Ask the user to input ten names and generate ten ***Person*** objects. Using the ***compareTo*** method, determine and the first and last person among them and print them.

***Modifications:***

Generate random names (first and last) from the data files provided. Also generate, for each person, a phone number. Make sure that these consist of 10 digits that create “legal” phone numbers. (Legal: An existing area code; second set of three digits does not start with zero; etc.)

The ***Person*** class should FirstName, LastName, and Phone attributes.

There may be duplicate names and/or phone numbers. (People have identical names; phone numbers may be shared.)

Using data sets of various sizes (1,000 to 1,000,000), use a timer to determine how quickly a name or phone number can be found. Write a brief report summarizing these comparisons.

***Big Bonus:***

A binary search has a Big-Oh of log(n) but requires a sorted list. You want to use a log(n) search because it is really fast. You do not want to keep duplicate lists, one sorted by name and one by phone number, as that would waste storage. You do not want to have to frequently re-sort a list if the look-ups alternate between name and number. How can you achieve this?